**Оглавление**

[Аннотация 4](#_Toc26196310)

[Введение 5](#_Toc26196311)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ 6](#_Toc26196312)

[1.1 Основное функциональное назначение программного средства 6](#_Toc26196313)ё

[1.2 Полное наименование программного средства 6](#_Toc26196314)

[1.3 Условное обозначение программного средства 6](#_Toc26196315)

[1.4 Разработчики программного средства 6](#_Toc26196316)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc26196317)

[2.1 Основание для разработки 7](#_Toc26196318)

[2.2 Назначение разработки 7](#_Toc26196319)

[2.3 Требование к программному средству 7](#_Toc26196320)

[2.4 Требования к программной документации 8](#_Toc26196321)

[2.5 Требования к эргономике и технической эстетике 8](#_Toc26196322)

[2.6 Стадии и этапы разработки 9](#_Toc26196323)

[2.7 Порядок контроля и приемки 10](#_Toc26196324)

[3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ 11](#_Toc26196325)

[3.1 Декомпозиция поставленной задачи 11](#_Toc26196326)

[3.2 Общая архитектура программного средства 12](#_Toc26196327)

[3.3 Разработка алгоритма решения задачи 15](#_Toc26196328)

[3.4 Реализация функционального назначения программного средства 16](#_Toc26196329)

[3.5 Структурная организация данных 16](#_Toc26196330)

[3.6 Разработка интерфейса ПС 20](#_Toc26196331)

[3.7 Описание структуры выходной информации 25](#_Toc26196332)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 26](#_Toc26196333)

[4.1 Назначение программного средства 26](#_Toc26196334)

[4.2 Условия выполнения программного средства 26](#_Toc26196335)

[4.3 Эксплуатация программного средства 26](#_Toc26196336)

[4.4 Сообщения пользователю 32](#_Toc26196337)

[5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА 34](#_Toc26196338)

[5.1 Обоснование необходимого количества тестов 34](#_Toc26196339)

[5.2 Описание тестовых пакетов 35](#_Toc26196340)

[6. ЗАКЛЮЧЕНИЕ 42](#_Toc26196341)

[6.1 Оценка качества программного средства с помощью метрик 42](#_Toc26196342)

[7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 45](#_Toc26196343)

[Приложение А листинг программного модуля 47](#_Toc26196344)

[Приложение B диск с исполняемым кодом программного модуля 65](#_Toc26196345)

**Аннотация**

На курсовой проект студента Медведева В.А. на тему: Разработка интегрированного программного модуля «Калькулятор подбора материалов для ремонта квартиры». Курсовой проект представлен пояснительной запиской на 65 листов машинописного текста. Состоит из введения, 5 глав, заключения и списка литературных источников.

В главе 1 содержится общие сведения о программном средстве Калькулятор подбора материалов для ремонта квартиры. В главе 2 приведено техническое задание на проектирование БД согласно ГОСТ 19.201. Глава 3 содержит пояснительную записку к программному продукту согласно ГОСТ 19.404-79. В главе 4 содержится руководство пользователя согласно ГОСТ 19.504-79. Глава 5 содержит обоснование количества необходимых тестов и тестовые пакеты для оценки работоспособности программного продукта. В главе 6 приведены результаты ручного и автоматического тестирования программного продукта и выполнена оценка качества с помощью метрик.

В работе содержится общее описание программного средства, предназначенного для объекта проектирования – интегрированного программного модуля «Калькулятор подбора материалов для ремонта квартиры», изложены требования к программному средству, описаны его архитектура, реализация функций, интерфейс, приведено руководство пользователя, обоснованы тестовые пакеты и выполнена оценка качества программного продукта с помощью метрик.

**Введение**

В курсовом проекте была поставлена задача разработки интегрированного программного модуля «Калькулятор подбора материалов для ремонта квартиры», который позволяет автоматизировать учет мероприятий.

Для достижения поставленной цели был составлен основной алгоритм работы программы, а также разработаны алгоритмы вспомогательных процедур.

Для реализации алгоритмов был выбран язык программирования C#. Выбор обусловлен тем, что возможностей этого языка достаточно для достижения поставленной цели моей курсового проекта.

При разработке интерфейса программы, я учитывал наиболее нужные и важные функциональные возможности, которые должна выполнять данная программа, а также тип пользователей, которые с ней будут работать.

**1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ**

**1.1 Основное функциональное назначение программного средства**

Программа предназначена для автоматизации учета мероприятий. Программа не требует владения современными информационными технологиями для эффективного использования данного программного средства, т.е. подходит для любого пользователя.

**1.2 Полное наименование программного средства**

Разработка интегрированного программного модуля «Калькулятор подбора материалов для ремонта квартиры».

**1.3 Условное обозначение программного средства**

«MaterialsCalculator1.01»

**1.4 Разработчики программного средства**

Программное средство разработала студент Зеленодольского Механического колледжа группы 205 Медведев В.А.

**2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1 Основание для разработки**

Работа ведётся на основании задания на курсовой проект по специальности 09.02.03 — «Программирование в компьютерных системах».

**2.2 Назначение разработки**

Программа «MaterialsCalculator1.01» работает под управление операционной системы Windows 7, Windows 8, Windows 10. Программа предназначена для автоматизации деятельности администратора по учету мероприятий города. Программа позволяет вводить, изменять и удалять данные мероприятиях, просматривать статусы.

**2.3 Требование к программному средству**

«MaterialsCalculator1.01» устанавливается на персональный компьютер, имеющий представленные минимальные системные требования:

Оперативная память: 4 Гб;

Процессор: Intel Core i / Phenom II X4 / AMD Athlon64 X2 4600 + 2.4 ГГц;

Видео карта: nVidia GeForce 8600 / ATI Radeon HD 2400;

Звуковая карта: Совместимая с DirectX;

Программный продукт «MaterialsCalculator1.01» является кроссплатформенным продуктом, что позволяет не ограничиваться в работе с определенной версией ОС.

«MaterialsCalculator1.01» написана на объектно-ориентированном языке программировании C# (си-шарп) в интегрированной среде разработки Visual Studio 2019. В качестве СУБД используется MS SQL. Программа не может подлежать сторонней оптимизации и редактированию программного кода.

**2.4 Требования к программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;

2. сроки выполнения работ;

3. программу и методику испытаний;

4. эксплуатационные инструкции пользователю;

**2.5 Требования к эргономике и технической эстетике**

Система обеспечивает удобный для пользователей системный интерфейс, отвечающий следующим требованиям:

- единый стиль оформления для пользовательских интерфейсов;

- удобная, интуитивно понятная навигация в интерфейсе пользователя;

- взаимодействие пользователя с системой осуществляется на двух языках по выбору (русский и английский), для наибольшего удобства

Пользовательские интерфейсы системы спроектированы и разработаны с применением единых принципов графического представления информации и организации доступа к функциональным возможностям и сервисам. Разработан графический дизайн пользовательских интерфейсов, цветовые, шрифтовые и композиционные решения для отображения текстов, изображений, таблиц, гиперссылок, управляющих и навигационных элементов (меню, кнопок, форм и т.п.).

Экранные формы спроектированы с учетом требований унификации:

- все экранные формы пользовательского интерфейса выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций использованы сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении унифицированы;

- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) реализованы одинаково для однотипных элементов.

**2.6 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на нее;

5. выбор языков программирования;

6. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с пунктом Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение порядка и методики испытаний;

2. проведение испытаний;

3. корректировка программы и программной документации по результатам испытаний.

**2.7 Порядок контроля и приемки**

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом. Сдача-приемка осуществляется комиссией, в состав которой входят представители ГАПОУ и работодатель. По результатам приемки подписывается акт приемочной комиссии.

В указанном подразделе, согласно таблице «Сроки выполнения работ» технического задания, будет описаны перечень участвующих организаций, место и сроки проведения работ, согласно п. 2.8 ГОСТ 34.602-89.

Порядок согласования и утверждения приемочной документации должен регламентироваться организационно-распорядительной документацией организации, принимающей участие в создании программного продукта. Согласно разделу «Приемка результатов разработки» ГОСТ 15.001-88 для согласования и утверждения приемочной документации создается приемочная комиссия (приказом).

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

**3. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ**

**3.1 Декомпозиция поставленной задачи**

Для достижения цели работы была поставлена следующая задача:

Создать модуль “MaterialsCalculator ” отвечающий за сохранение БД. Результат работы модуля - сохранение БД.

Поставленные задачи наглядно отображаются на диаграмме деятельности.

Изображение выглядит как текст, диаграмма, План, линия

Автоматически созданное описание

Рис.1 Декомпозиция поставленной задачи

**3.2 Общая архитектура программного средства**

Объектом моделирования является Калькулятор подбора материалов для ремонта квартиры.

Рассматриваемые процессы: Добавление и удаление информации о мероприятиях.

Объекты моделирования представлены на диаграмме классов, рис.2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.2 Диаграмма классов

По полученной декомпозиции задач была спроектирована следующая архитектура программного средства.

Изображение выглядит как диаграмма, Технический чертеж, План, схематичный

Автоматически созданное описание

Рис.3 Схема событийно-ориентированной архитектуры

В «MaterialsCalculator1.01» используется событийно-ориентированная архитектура.(Рис.3) Роль агента (источник событий) в программе выполняют: пользователи программы, за роль стоков (потребители событий) отвечают таблицы, входящие в базу данных. Например, когда пользователь выбирает определенное действие: добавить, удалить, редактировать, вывести в Excel и т.д. система осуществит выбранные действия, и база отреагирует соответствующим образом: запись добавлена, удалена, отредактирована, либо отчет выведен в Excel. Системная архитектура пользователя рассматривает это изменение состояния как событие, создаваемое, публикуемое, определяемое и потребляемое различными приложениями в составе архитектуры.

Изображение выглядит как снимок экрана, диаграмма, дизайн

Автоматически созданное описание

Рис.4 Диаграмма вариантов использования

**3.3 Разработка алгоритма решения задачи**

**Общий алгоритм**

****Рис.5 Алгоритм программы

**3.4 Реализация функционального назначения программного средства**

Программа «MaterialsCalculator1.01» имеет следующий набор входных данных, такие как: материалы и информация о них.

Данные вводятся администратором в соответствующие поля ввода, снабженные всплывающими подсказками.

Выходными данными являются: отображение списка мероприятий в виде плиточного интерфейса.

Данные вводятся пользователем в соответствующие поля вывода, снабженные всплывающими подсказками.

Выходные данные редактировать вручную невозможно, т.к. они служат в качестве отчетной информации для пользователя.

**3.5 Структурная организация данных**

Для создания БД необходимо определиться с данными, которые необходимы для полноценного функционирования системы. Все эти данные указаны в реляционной модели «БД Калькулятор подбора материалов для ремонта квартиры» представленная на рисунке. Любая реляционная база данных и называется реляционной, что характеризуется отношениями (relation) между таблицами. На рисунке изображены таблицы моей базы данных. При этом одна таблица является родительской (главной), а вторая – дочерней (подчиненной). Главной таблицей является «Calculation». Реляционная модель автоматизированной системы соответствует всем 12 правилам Кодда. (рис.6)

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

Рис.6 Реляционная модель

Первичный ключ в базе уникален, используется для организации отношений между таблицами, который не может иметь пустых и повторяющихся значений. Первичными ключами в базе являются поля: Код\_расчета (таблица Culculation), Код\_материала (Таблица Material), Код\_юзера (Таблица Users), Код\_роли (таблица Role), Код\_пола (таблица Gender), Код\_типа\_материала (таблица MaterialType), Код\_производителя (таблица Manufacturer), Остальные ID-ключи являются внешними ключами.

Для организации более эффективной обработки данных применяется нормализация. Таблицы моей БД находятся в 3НФ:

* БД находится в форме -1НФ потому, что
  + Таблица не имеет повторяющихся записей;
  + Каждый атрибут отношения хранит одно-единственное значение и не является списком, ни множеством значений;
  + Таблица не имеет повторяющихся групп полей.
    - Вторая нормальная форма(2НФ):
      * Устранены атрибуты, зависящие только от части уникального (первичного) идентификатора, т.е. ID.
        + Третья нормальная форма(3НФ):

Отсутствуют атрибуты, зависящие от атрибутов, не входящих в уникальный (первичный) ключ.

На основе реляционной модели базы данных мною в MS SQL была построена следующая база данных. (Рис.7) В ней первичным ключом является поле таблицы “Calculation” – “CalculationID”. Связь таблиц “Material” и “Users” по полю “MaterialID” и по полю “UsersID” один ко многим (1:М). Таблицы “Manufacturer” и “MaterialType” связаны по полю “ManufacturerID” и “ MaterialTypeID” типом связи 1:М. Таблицы “Gender” и “Role” связаны по полю “GenderID” и “ RoleID” типом связи 1:М. Все связи между таблицами базы данных типа один ко многим.

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание

Рис.7 Схема базы данных

Таблица “Calculation ”(Рассчеты)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.8 Таблица Calculation

Таблица “Gender”(Пол)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.9 Таблица Gender

Таблица “Manufacturer”(Производитель)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.10 Таблица Manufacturer

Таблица “Material”(Материал)



Рис.11 Таблица Manufacturer

Таблица “MaterialType”(Тип материала)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.12 Таблица Manufacturer

Таблица “Role”(Роль)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.13 Таблица Manufacturer

Таблица “Manufacturer”(Производитель)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.14 Таблица Manufacturer

**3.6 Разработка интерфейса ПС**

В приложении предусмотрена авторизация, так как оно огопользовательское. Для запуска приложения достаточно щелкнуть дважды левой кнопкой мыши по ярлыку.

После старта приложения пользователь приступит к работе с главной формой программы, на которой и будут отображаться все события в базе данных. «MaterialsCalculator1.01» обладает удобным интерфейсом, так что даже неопытному в обращение с пользовательскими компьютерными программами пользователю не составит труда быстро привыкнуть к работе с данным программным средством.

На рабочем экране расположены несколько кнопок для комфортной работы с приложением.

**Страница Типов Материалов**. Основная форма приложения, в которой в виде плитки отображаются типы материалов. Компоненты формы: ListView – для отображения типов материалов, TextBox и Button. (Рис.15)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

TextBox

Button

ListView

Рис.15 Каталог

**Страница Материалов**. Компоненты Страницы: DataGrid – для отображения списка мероприятий, ListView – для отображения материалов, и кнопки Button для перехода на дополнительные страницы. (Рис.16)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Image

Button

TextBox

ListView

Рис.16 Калькулятор подбора материалов для ремонта квартиры

**Страница Добавление**. Интерфейс данной формы состоит из компонентов: TextBox – поля ввода, ComboBox – выпадающие списки, Image – дляотображения изображения и Button. Также есть возможность открытия дополнительных окон. (Рис.17)

**Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание**

TextBox

Button

ComboBox

Image

Рис.17 Страница Добавление и редактирование

**Страница Редактирование**. Интерфейс данной формы состоит из компонентов: TextBox – поля ввода, ComboBox – выпадающие списки, Image – для отображения изображения и Button. Также есть возможность открытия дополнительных окон. (Рис.17)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

TextBox

Button

ComboBox

Image

Рис.18 Страница Категории

**Страница Юзера**. Интерфейс данной формы состоит из компонентов: Image – для отображения изображения, ListView – для отображения расчетов, DataGrid и кнопок.(Рис.15)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Button

Image

Рис.19 Страница Статусы

ListView

**Страница Админа**. Интерфейс данной формы состоит из компонентов: кнопок.(Рис.15)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Button

Рис.20 Страница Статусы

**3.7 Описание структуры выходной информации**

Выходными данными являются списки иатериалов отфильтрованные по названию.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

TextBox

Button

ListView

Рис.21 Каталог

**4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Назначение программного средства**

Назначением программного средства является автоматизация учета мероприятий.

**4.2 Условия выполнения программного средства**

Условия, при соблюдении которых возможно выполнение работы с программным продуктом:

-Компьютер с установленной на нём «MaterialsCalculator1.01».

Подготовительные действия:

-На иконке «MaterialsCalculator1.01» рабочего стола произвести двойной щелчок левой кнопкой мыши.

**4.3 Эксплуатация программного средства**

Этот раздел содержит обзор приложений «MaterialsCalculator1.01» и ее основных параметров.

***Активация продукта***

Активировать копию программы MaterialsCalculator1.01 можно сразу после её установки, когда появится окно подтверждения подлинности Вашей версии. Ключ активации расположен внутри упаковки программного продукта. Для успешного выполнения активации ключ активации необходимо вводить именно в том виде, в котором он предоставлен.

Вводные сведения об интерфейсе пользователя

После старта приложения откроется главное окно приложения – Страница типов материалов в списочном интерфейсе. На форме есть кнопка для перехода на страницу материалов, регистрации/авторизации и страницу пользователя. Есть поля для поиска мероприятия по назвванию. (Рис.22)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.22 Окно Калькулятор подбора материалов для ремонта квартиры

Кликнув по кнопке «Выход» программа потребует подтверждение. (Рис.23)

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Рис.23 Сообщение программы

Ниже описаны пункты меню управления и информации.

«Страница Материалов»: эта форма предназначена для отображения списка материалов, а также для расчета ихцены. (Рис.19)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис.24 Страница Материалы

«Страница Добавление»: эта форма предназначена для добавление данных в таблице Материал. (Рис.22)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.25 Страница Добавление и редактирование

«Страница Добавление»: эта форма предназначена для редактирования данных в таблице Материал. (Рис.26)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.26 Страница Редактирование Материалов

«Админ»: эта страница предназначена для добавления, удаления и редактирования данных в таблице Материал. (Рис.28)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.27 Страница Админ

«Юзер»: эта страница предназначена для просмотре результатов расчетов. (Рис.28)

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис.28 Страница Юзер

Работа с MaterialsCalculator1.01

После запуска программы Вы перейдете на главную форму программы. В формах «Авторизация», «Регистрация», «Материалы», «Добавление материала», «Редактирование материала», «Удаление материала» заполняем все необходимые для работы пункты.

После этого кликаем по кнопке «ок».

Если Вы желаете удалить любую введенную ранее информацию, Вам необходимо кликнуть по нужной кнопке.

Для того чтобы изменить запись, введите в редактируемые поля данные и нажмите кнопку сохранить. Данные формы редактированию и управлению со стороны пользователя не подлежат.

После того, как Вы закончите сеанс работы с программой, если Вы хотите выйти из MaterialsCalculator1.01 кликните на соответствующей кнопке «Выход».

Основные кнопки MaterialsCalculator1.01

Кнопки управления «Добавить запись», «Удалить запись», «Сохранить» расположены в соответствуюзих местах.

Действия этих кнопок приведены в таблице 1:

Таблица-1 Действие кнопок

|  |  |
| --- | --- |
| Добавить запись | Прежде чем добавить запись в таблицу необходимо нажать на данную кнопку. Поля формы будут очищены, пользователю требуется ввести необходимые данные. Если пользователь нажмет кнопку «Сохранить данные», то введенная информация добавится в БД. Если нажать кнопку «Отменить», то введенная информация пользователем удалится. |
| Удалить запись/отменить | Кнопка «Удалить запись» позволяет отменить действия при вводе, либо удаляет конкретную запись. |
| Сохранить данные/обновить запись | Кнопка «Сохранить данные» добавляет запись введенную пользователем в БД, либо обновляет существующую. |
| Редактировать | Открывает форму в режиме редактирования информации |
|  | «Поиск» позволяет найти нужную информацию. |

**4.4 Сообщения пользователю**

Для уверенной и комфортной работы пользователю важно понимать поведение программы, а программе, в свою очередь, поведение пользователя. Для информирования и уточнения намерений пользователей в программах используются сообщения.

При попытке удалить запись, пользователю выводится сообщение с запросом на подтверждение или отмену действия. (Рис.23)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рис.23 Сообщение Подтверждение удаления

После добавления новой записи, пользователю выводится сообщение. (Рис.24)

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.24 Сообщение Запись добавлена

После обновления записи, пользователю выводится сообщение. (Рис.25)



Рис.25 Сообщение Запись обновлена

**5. РАЗРАБОТКА ТЕСТОВОГО НАБОРА**

**5.1 Обоснование необходимого количества тестов**

Для разработанного алгоритма был построен граф.

****

Рис.27 Потоковый граф

Необходимое количество тестов определяется по формуле V(G) = E-N+2, где E – это число ребер, а N–это число вершин данного графа. Согласно представленному графу E=34, N=25. Далее по формуле рассчитаем необходимое количество тестовых сценариев (тест-кейсов).

V(G)=34-25+2=11.

Для данного программного средства необходимо разработать не менее 11 тестовых сценариев.

Чтобы определить количество тестовых прогонов для программного продукта MaterialsCalculator1.01 необходимо выделить требования. Из требований к приложению выделим поддержку 2-х операционных систем с одним основным языком локализации и выполнения 6-ти основных функций. Кроме того, приложение позволяет выполнять 5 функций по взаимодействию с системой (запуск приложения, выход из приложения, сохранение результатов в файл, работа с буфером и т.п.).

Таким образом, полное покрытие требований задаёт набор из 2\*1\*(6+5) =22 тестовых прогонов.

**5.2 Описание тестовых пакетов**

Таблица 1 – Тестирование корректной авторизации пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректной авторизации пользователя | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален |
| **Предусловие:** | |  |  |
| Открываем форму авторизации | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Заполните форму  Почта = qwe@qwe.com  Пароль = Qwerty!1 | | Данные успешно введены |  |
| Нажмите кнопку «Войти» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма основное окно открыта | | Отображается форма основное окно |  |

Таблица 2 – Тестирование некорректной авторизации пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректной авторизации пользователя | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален |
| **Предусловие:** | |  |  |
| Открываем форму авторизации | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Заполните форму  Почта = 1  Пароль = 1 | | Сооюзение о некорректоном формате полей |  |
| Нажмите кнопку «Войти» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма основное окно открыта | | Отображается форма авторизации |  |

Таблица 3 – Тестирование корректного добавления информации в Материал

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Материалы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = 1  Описание = красный  Фото=robin.png  Тип материала=Tile  Производитель=TileCraft  Площадь=1  Цена=1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Добавление материала открыта | | Отображается форма Добавление материала |  |

Таблица 4 – Тестирование некорректного добавления информации в таблицу Материал

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Материалы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = 1  Описание = красный  Фото=robin.png  Тип материала=Tile  Производитель=TileCraft  Площадь=йцу  Цена=йцу | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение о неверном формате полей | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Добавление материала открыта | | Отображается форма Добавление материала |  |

Таблица 5 – Тестирование корректного редактирования информации в таблице Материалы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Материалы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму редактирование материала | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = 2  Описание = красный  Фото=robin.gif  Тип материала=qwe  Производитель=qweqw  Площадь=qwe  Цена=qwe | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообзение о некорректных полях | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма редактирование материала открыта | | Отображается форма редактирование материала |  |

Таблица 6 – Тестирование некорректного редактирования информации в таблице Материалы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Материалы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму редактирование материала | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = 2  Описание = красный  Фото=robin.png  Тип материала=Tile  Производитель=TileCraft  Площадь=йцу  Цена=йцу | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение о неверном формате полей | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма редактирование материала открыта | | Отображается форма редактирование материала |  |

Таблица 9 – Тестирование корректного добавления информации в таблицу Юзер

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Мероприятие | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Регистрация | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фамилия= 1  Имя =1  Отчество=1  Почта=1@1.com  Пароль=Qwerty!1  Повтор пароля= Qwerty!1 | | Данные успешно введены |  |
| Нажмите кнопку «Ок» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Страница пользователя открыта | | Отображается страница пользователя |  |

Таблица 8 – Тестирование некорректного добавления информации в таблицу Юзер

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Мероприятие | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Регистрация | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фамилия= 1  Имя =1  Отчество=1  Почта=1@1.com  Пароль=Qwerty!1  Повтор пароля= Qwerty!1 | | Данные успешно введены |  |
| Нажмите кнопку «Ок» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Страница пользователя открыта | | Отображается страница пользователя |  |

Таблица 9 – Тестирование некорректного добавления информации в таблицу Culculator

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Мероприятие | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Мероприятие | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку выбрать  Заполните форму  Площадь=1йццуу  Нажмите кнопку расчитать | | Данные успешно введены |  |
| Нажмите кнопку «Ок» | | Сообщение о некорректных полях | **Тест пройден** |
| **Постусловие:** | |  |  |
| Отображается страница Материалы | | Отображается форма Материалы |  |

Таблица 10 – Тестирование некорректного добавления информации в таблицу Culculator

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Мероприятие | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Отображается страница Материалы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку выбрать  Заполните форму  Площадь=100  Нажмите кнопку расчитать | | Данные успешно введены |  |
| Нажмите кнопку «Ок» | | Сообзение о некорректных полях | **Тест пройден** |
| **Постусловие:** | |  |  |
| Отображается страница Материалы | | Отображается форма Материалы |  |

Таблица 11 – Тестирование некорректного удаления информации в таблице Материал

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Мероприятие | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Материалы для удаления | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите удалить  Поддтвердите удаление во всплывающем окне | | Данные успешно введены |  |
| Нажмите кнопку «Ок» | | Сообщение выюреите материал для удаления | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Материалы для удаления открыта | | Отображается форма Материалы для удаления |  |

Таблица 12 – Тестирование корректного удаления информации в Материал

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Категории | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Материалы для удаления | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для удаления и нажмите удалить  Поддтвердите удаление во всплывающем окне | | Данные успешно удалены |  |
| Нажмите кнопку «Ок» | | Данные успешно удалены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Материалы для удаления открыта | | Отображается форма Материалы для удаления |  |

**6. ЗАКЛЮЧЕНИЕ**

**6.1 Оценка качества программного средства с помощью метрик**

Оценка качества программного средства проводится исходя из стандарта оценки качества ISО 9126.

На компьютере следующей конфигурации:

Установленная операционная система – Windows 7 Ultimate 32бит.

Процессор: Intel Pentium G2030 3.0 GHz.

ОЗУ: 2 Gb

Таблица 14 – Требования к количественным характеристикам качества программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Надежность** |  | | |
| *Завершенность:*  *-* наработка на отказ при отсутствии рестарта; | Часы | 10 | 10 |
| *Устойчивость:*  - относительные ресурсы на обеспечение надежности и рестарта. | % | 0,5 | 0,5 |
| *Восстанавливаемость:*  -длительность восстановления. | Минуты | 2 | 2 |
| *Доступность-готовность:*  -относительное время работоспособного функционирования. | Вероятность | 0,998 | 0,998 |
| **Эффективность** |  | | |
| *Временная эффективность:*  *-*время отклика - получение результатов на типовое задание;  - пропускная способность число типовых заданий, исполняемых в единицу времени. | Секунды  Число в минуту | 3  60 | 3  60 |
| *Используемость ресурсов:*  *-* относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства. |  | 0,001 | 0,001 |

Таблица 15 – Требования к качественным характеристикам программного средства.

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Практичность** |  | | |
| *Простота использования:*  **-**среднее время ввода заданий;  -среднее время отклика на задание. | Секунды  Секунды | 10  3 | 10  3 |
| *Изучаемость :*  - трудоемкость изучения применения ПС;  -продолжительность изучения;  -объем эксплуатационной документации; | Чел.-часы  Часы  Страницы | 5  2  15 | 5  2  15 |
| **Сопровождаемость** |  | | |
| *Изменяемость:*  -трудоемкость подготовки изменений;  -длительность подготовки изменений. | Чел.-часы  Часы | 5  5 | 5  5 |
| *Тестируемость:*  *-трудоемкость тестирования изменений;*  *- длительность тестирования изменений.* | Чел.-часы  Часы | 3  3 | 3  3 |
| **Мобильность** |  | | |
| *Адаптируемость:*  - трудоемкость адаптации;  -длительность адаптации. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Простота установки:*  -трудоемкость инсталляции;  -длительность инсталляции. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Замещаемость:*  -трудоемкость замены компонентов;  - длительность замены компонентов. | Чел.-часы  Часы | 0,2  0,2 | 0,2  0,2 |

Таблица 16 – Сложность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип объекта | Количество | Вес | | | Итого |
|  |  | Простой | Средний | Сложный |  |
| Экран | 9 | x1 | x2 | **x3** | =27 |
| Отчет | 2 | x2 | **x5** | x8 | =10 |

Таблица 17 – Сложность экрана

|  |  |  |  |
| --- | --- | --- | --- |
| Экраны | Количество Категорииских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | **Всего <8** | Всего >8 |
|  | (<2 срв, <3 клт) | **(2-3 срв, 3-5 клт)** | (>3 срв, >5 клт) |
| <3 | Простой | Простой | Средний |
| 3-7 | Простой | Средний | Сложный |
| **>8** | Средний | **Сложный** | Сложный |

Таблица 18 – Сложность отчета

|  |  |  |  |
| --- | --- | --- | --- |
| Отчеты | Количество Категорииских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | Всего >8 |
|  | (<2 срв, <3 клт) | **(2-3 срв, 3-5 клт)** | (>3 срв, >5 клт) |
| 0 или 1 | Простой | Простой | Средний |
| **2 или 3** | Простой | **Средний** | Сложный |
| >4 | Средний | Сложный | Сложный |

Производительность = 4 тысяч строк кода/2 месяца= 2 тысяч строк кода/месяц

Качество = 2 ошибки /4 тысяч строк кода=0,5 ошибок/тысяч строк кода

**7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Г. Шилдт C#: Учебный курс. - Спб.: Питер, 2021. - 512с

2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2021. - 684 с.

3. Ватсон К. Си Шарп/К.Ватсон.-М.:Лори, 2022.-862 с

4. Балена Ф., Димауро Д. Современная практика программирования на Visual Basic и Visual C#.-М.:Русская редакция,2021,604с.

5. Енин А., Енин Н. Локальная СУБД своими руками. Учимся на примерах.-М.:СОЛОН-ПРЕСС,2022,464с.,СД

6. Лабор В.В. Си Шарп: СоЗдание приложений для Windows/-Мн.: Харвест,2022-384с.

7. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2022-512с.

8. ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению

9. ГОСТ 19.102-77.ЕСПД. Стадии разработки

10. ГОСТ 19.505-79.ЕСПД. Руководство оператора. Требования к содержанию и оформлению

11. ГОСТ 19.002-80.ЕСПД. Схемы алгоритмов и программ. Правила выполнения

12. ГОСТ 19.404-79.ЕСПД. Пояснительная записка. Требования к содержанию и оформлению

13. Владимир Биллинг. Основы программирования на С#. 2021 г.

[Электронный ресурс http://mexalib.com/view/42671]

14. Никита Культин. Основы программирования в MicrosoftVisualC# 2022. 2023 г.

[Электронный ресурс http://www.litres.ru/nikita-kultin/osnovy-programmirovaniya-v-microsoft-visual-c-2022-2]

15. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2021.

[Электронный ресурс http://www.twirpx.com/file/27852]

16. Йен Гриффитс. Программирование на С#. 2021 г.

[Электронный ресурс http://rutracker.org/forum/viewtopic.php?t=4772853]

**Приложение А**

Разработка интегрированного программного модуля «Калькулятор подбора материалов для ремонта квартиры»

Текст кода программного средства

460.ЗМК.19-01 12 01

**Текст кода программного средства**

**Словарь ресурсов**

<Application x:Class="MaterialsCalculator.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:MaterialsCalculator"

StartupUri="Windows\MainWindow.xaml">

<Application.Resources>

<!--цвета приложения для соответсвия стилю-->

<SolidColorBrush x:Key="main" Color="#DCDCDC "/>

<SolidColorBrush x:Key="additional" Color="#4D4D4D"/>

<SolidColorBrush x:Key="akcent" Color="#FF4D00"/>

<!--Стиль для окна-->

<Style TargetType="Window" x:Key="base\_window">

<Setter Property="FontFamily" Value="Nunito Gothic"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="MinHeight" Value="600"/>

<Setter Property="MinWidth" Value="800"/>

<Setter Property="Background" Value="{StaticResource main}"/>

</Style>

<!--Стиль для страницы-->

<Style TargetType="Page" x:Key="base\_page">

<Setter Property="FontFamily" Value="Nunito Gothic"/>

<Setter Property="Background" Value="{StaticResource main}"/>

<Setter Property="FontSize" Value="14"/>

</Style>

<!--Стиль для полей ввода TextBox-->

<Style TargetType="TextBox">

<Setter Property="Height" Value="30"/>

<Setter Property="VerticalAlignment" Value="Stretch"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Padding" Value="3 0"/>

<Setter Property="Background" Value="{StaticResource main}"/>

</Style>

<!--Стиль для полей кнопок Button-->

<Style TargetType="Button">

<Setter Property="Width" Value="auto"/>

<Setter Property="Height" Value="auto"/>

<Setter Property="Background" Value="{StaticResource akcent}"/>

<Setter Property="Foreground" Value="{StaticResource main}"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Margin" Value="3"/>

<Setter Property="Padding" Value="5"/>

<Setter Property="Height" Value="30"/>

</Style>

<!--Стиль для меток TextBlock -->

<Style TargetType="TextBlock" x:Key="base\_textblock">

<Setter Property="Height" Value="30"/>

<Setter Property="VerticalAlignment" Value="Stretch"/>

<Setter Property="HorizontalAlignment" Value="Stretch"/>

<Setter Property="Padding" Value="5"/>

<Setter Property="Foreground" Value="{StaticResource main}"/>

</Style>

<Style TargetType="TextBlock" x:Key="add\_window\_textblock">

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Right"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Margin" Value="0 0 10 0"/>

</Style>

<!--Стиль для меток TextBlock -->

<Style TargetType="TextBlock" x:Key="item\_textblock">

<Setter Property="TextAlignment" Value="Center"/>

<Setter Property="VerticalAlignment" Value="Top"/>

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Padding" Value="3"/>

</Style>

<!--Стиль для меток ListViewItem -->

<Style TargetType="ListBoxItem" x:Key="good\_item">

<Setter Property="Background" Value="{Binding GetColor}" />

<Setter Property="BorderBrush" Value="{StaticResource akcent}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="HorizontalContentAlignment" Value="Stretch"/>

</Style>

</Application.Resources>

</Application>

**Класс Manager**

using System.Windows.Controls;

namespace MaterialsCalculator.Models

{

public class Manager

{

/// <summary>

/// Фрейм, в котором отбражаются Page

/// </summary>

public static Frame MainFrame { get; set; }

/// <summary>

/// Текущий пользователь системы

/// </summary>

public static User CurrentUser { get; set; }

}

}

**Класс** MaterialType**.cs**

using System;

using System.IO;

namespace MaterialsCalculator.Models

{

public partial class MaterialType

{

public string GetPicture

{

get

{

if (Picture is null)

return Directory.GetCurrentDirectory() + @"\Images\picture.png";

return Directory.GetCurrentDirectory() + @"\Images\MaterialTypes\" + Picture.Trim();

}

}

}

}

**Класс** MaterialsCalculatorEntities**.cs**

using System.Data.Entity;

namespace MaterialsCalculator.Models

{

public partial class MaterialsCalculatorEntities: DbContext

{

private static MaterialsCalculatorEntities \_context;

/// <summary>

/// Метод возвращающий контекст подключения

/// </summary>

/// <returns></returns>

public static MaterialsCalculatorEntities GetContext()

{

if (\_context == null)

{

\_context = new MaterialsCalculatorEntities();

}

return \_context;

}

}

}

**Класс** MakeCaptcha**.cs**

using System;

using System.Drawing;

using System.IO;

using System.Windows.Media.Imaging;

namespace MaterialsCalculator.Models

{

public class MakeCaptcha

{

/// <summary>

/// Генерирует картинку, содержащую текст капчи

/// </summary>

/// <param name="width">ширина изображения</param>

/// <param name="height">высота изображения</param>

/// <param name="symbolCount">длина капчи</param>

/// <returns>возвращает картинку и текст капчи</returns>

public static (BitmapImage image, string captcha) CreateImage(int width, int height,

int symbolCount)

{

Random rnd = new Random();

//Создадим изображение

Bitmap result = new Bitmap(width, height);

//Добавим различные цвета

Brush[] colors = { Brushes.Black,

Brushes.Red,

Brushes.RoyalBlue,

Brushes.Green };

//Укажем где рисовать

Graphics g = Graphics.FromImage((System.Drawing.Image)result);

//Пусть фон картинки будет серым

g.Clear(Color.Gray);

//Сгенерируем текст

char symbol;

// нбор допустимых символов

string alphabet = "1234567890QWERTYUIOPASDFGHJKLZXCVBNM";

// текст капчи

string captcha = "";

// размер поля для одного символа

int h = width / symbolCount;

for (int i = 0; i < symbolCount; ++i)

{

// генерируем размер буквы

int size = rnd.Next(20, h);

// выбираем любой символ из допустимого набора

symbol = alphabet[rnd.Next(alphabet.Length)];

// текст капчи

captcha += symbol;

// генерируем позиции рисования символа

int Ypos = rnd.Next(height - size - 20);

int Xpos = rnd.Next(i \* h, (i + 1) \* h - size);

//Нарисуем сгенерированный символ

g.DrawString(symbol.ToString(),

new Font("Arial", size),

colors[rnd.Next(colors.Length)],

new PointF(Xpos, Ypos));

//Добавим немного помех

///////Линии из углов

//Устанавливаем цвет линии

Pen skyBluePen = new Pen(colors[rnd.Next(colors.Length)]);

// Устанавливаем толщину линии

skyBluePen.Width = rnd.Next(2, 5);

g.DrawLine(skyBluePen,

new Point(Xpos + 5, Ypos + 5),

new Point(Xpos + size + 5, Ypos + size + 5));

}

////Белые точки

for (int i = 0; i < width; ++i)

for (int j = 0; j < height; ++j)

if (rnd.Next() % 20 == 0)

result.SetPixel(i, j, Color.White);

return (BitmapToImageSource(result), captcha);

}

/// <summary>

/// Конвертирует Bitmap в BitmapImage для использования в качестве источника изображения для Image

/// </summary>

/// <param name="bitmap">Картинка созданная программным образом</param>

/// <returns>BitmapImage, который можно использовать в качестве ImageSource для

/// Image</returns>

private static BitmapImage BitmapToImageSource(Bitmap bitmap)

{

using (MemoryStream memory = new MemoryStream())

{

bitmap.Save(memory, System.Drawing.Imaging.ImageFormat.Bmp);

memory.Position = 0;

BitmapImage bitmapimage = new BitmapImage();

bitmapimage.BeginInit();

bitmapimage.StreamSource = memory;

bitmapimage.CacheOption = BitmapCacheOption.OnLoad;

bitmapimage.EndInit();

return bitmapimage;

}

}

}

}

Класс Manager

using System.Windows.Controls;

namespace MaterialsCalculator.Models

{

public class Manager

{

/// <summary>

/// Фрейм, в котором отбражаются Page

/// </summary>

public static Frame MainFrame { get; set; }

/// <summary>

/// Текущий пользователь системы

/// </summary>

public static User CurrentUser { get; set; }

}

}

Окно админа

<Page x:Class="MaterialsCalculator.Pages.AdminPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Pages"

mc:Ignorable="d"

Style="{StaticResource base\_page}"

Height="450" Width="800"

Title="AdminPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}"/>

<Image Margin="10"

HorizontalAlignment="Left"/>

<TextBlock Grid.Row="0" Text="Окно админа"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20"

FontWeight="Bold"/>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="300"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="1">

<TextBlock x:Name="TextBlockUserName" Height="30"

Foreground="{StaticResource additional}"/>

<Button Content="Добавить материал" Click="Button\_Click"/>

<Button Content="Удалить / Изменить материял" Click="Button\_Click\_1"/>

</StackPanel>

</Grid>

<Grid Grid.Row="2" Background="{StaticResource additional}"/>

</Grid>

</Page>

using System;

using System.Windows.Controls;

using MaterialsCalculator.Models;

using MaterialsCalculator.Windows;

using System.Windows;

using System.Windows.Media.Imaging;

using System.Windows.Threading;

namespace MaterialsCalculator.Pages

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public User user;

public AdminPage(User user)

{

InitializeComponent();

this.user = user;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

AddMaterialWindow addWindow = new AddMaterialWindow();

addWindow.ShowDialog();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

DeleteMaterialsWindow delWindow = new DeleteMaterialsWindow();

delWindow.ShowDialog();

}

}

}

Окно производителя

<Page x:Class="MaterialsCalculator.Pages.ManufacturerPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ManufacturerPage"

IsVisibleChanged="Page\_IsVisibleChanged"

Style="{StaticResource base\_page}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch"

Background="{StaticResource additional}" Grid.ColumnSpan="2">

</WrapPanel>

<StackPanel Margin="10" Grid.Row="1">

<TextBlock x:Name="TBName" Text="{Binding Manufacturer.Name}" FontSize="18" FontWeight="Bold" Margin="0,0,0,10"/>

<TextBlock x:Name="TBDeccription" Text="{Binding Manufacturer.Description}" Margin="0,0,0,5"/>

<TextBlock x:Name="TBEmail" Text="{Binding Manufacturer.Email}" Margin="0,0,0,5"/>

</StackPanel>

</Grid>

</Page>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace MaterialsCalculator.Pages

{

/// <summary>

/// Логика взаимодействия для ManufacturerPage.xaml

/// </summary>

public partial class ManufacturerPage : Page

{

public Material material { get; }

public ManufacturerPage(Material material)

{

InitializeComponent();

this.material = material;

this.LoadData();

}

public void LoadData()

{

try

{

Manufacturer manufacturer = MaterialsCalculatorEntities.GetContext().Manufacturers.ToList().Where(p => p.ManufacturerID == material.ManufacturerID).FirstOrDefault();

TBName.Text = $"Manufacturer Name: {manufacturer.Name}";

TBDeccription.Text = $"Description: {manufacturer.Description}";

TBEmail.Text = $"Email: {manufacturer.Email}";

}

catch (NullReferenceException ex)

{

MessageBox.Show("Похоже, что-то пошло не так :-(" + ex.Message);

TBName.Text = $"Manufacturer Name: 404";

TBDeccription.Text = $"Description: 404";

TBEmail.Text = $"Email: 404";

}

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//обновление данных после каждой активации окна

if (Visibility == Visibility.Visible)

{

MaterialsCalculatorEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

}

}

}

}

Окно материалов

<Page x:Class="MaterialsCalculator.Pages.MaterialPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Мероприятия"

IsVisibleChanged="Page\_IsVisibleChanged"

Style="{StaticResource base\_page}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="60"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="60"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="450"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch"

Background="{StaticResource additional}" Grid.ColumnSpan="2">

</WrapPanel>

<ListBox x:Name="ListBoxMaterials"

ScrollViewer.VerticalScrollBarVisibility="Visible"

SelectedValuePath="AgentId"

Grid.Row="1" Grid.Column="0" Grid.RowSpan="4">

<ListBox.ItemTemplate>

<DataTemplate >

<Grid HorizontalAlignment="Stretch" UseLayoutRounding="True">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Image Source="{Binding Path=GetImage}" Grid.Row="0" Stretch="UniformToFill" HorizontalAlignment="Left" Grid.RowSpan="2">

</Image>

<TextBlock

Text="{Binding Path=Name}"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="20" Grid.Row="0" Grid.Column="1" />

<TextBlock

Text="{Binding Path=Description}"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="15" Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="2"/>

<TextBlock

Text="{Binding Path=Square}"

VerticalAlignment="center" TextAlignment="Center" Width="100" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="15" Grid.Row="2" Grid.Column="1" />

<TextBlock

Text="{Binding Path=Price}"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="15" Grid.Row="2" Grid.Column="3"/>

<TextBlock

Text="Площадь"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Center" FontSize="15" Grid.Row="2" Grid.Column="0"/>

<TextBlock

Text="Цена"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Center" FontSize="15" Grid.Row="2" Grid.Column="2"/>

<Button Content="Производитель" Click="Button\_Click" Grid.Column="3" Grid.Row="0"

HorizontalAlignment="Right" Width="100">

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

<Button Content="Выбрать" Click="Button\_Click\_2" Grid.Column="3" Grid.Row="1"

HorizontalAlignment="Right" Width="100">

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<TextBlock Width="100" Height="25" Grid.Column="2" Grid.Row="1" Text="Площадь" HorizontalAlignment="Left" Margin="40 0 10 0" TextAlignment="Center"/>

<TextBlock/>

<TextBox x:Name="TBoxSquare" Width="120" Grid.Column="1" Grid.Row="1" HorizontalAlignment="Right" Margin="0 0 30 0"/>

<Button Content="Расчитать" Click="Button\_Click\_1" Grid.Column="2" Grid.Row="2" Margin="40 0 10 0" HorizontalAlignment="Left" Width="120" >

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

<TextBox IsReadOnly="True" x:Name="Result" Width="120" Grid.Column="2" Grid.Row="2" Margin=" 0 0 30 0" HorizontalAlignment="Right"/>

</Grid>

</Page>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System;

namespace MaterialsCalculator.Pages

{

/// <summary>

/// Логика взаимодействия для MaterialPage.xaml

/// </summary>

public partial class MaterialPage : Page

{

public MaterialType materialType;

public Material \_curentMaterial;

public MaterialPage(MaterialType materialType)

{

InitializeComponent();

this.materialType = materialType;

LoadAndInitData();

}

void LoadAndInitData()

{

// загрузка мероприятий в ListBox сортируем по Дате

var mce = MaterialsCalculatorEntities.GetContext().Materials.ToList().Where(p => p.MaterialTypeID==materialType.MaterialTypeID);

ListBoxMaterials.ItemsSource = mce;

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//обновление данных после каждой активации окна

if (Visibility == Visibility.Visible)

{

MaterialsCalculatorEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (sender is Button button && button.Tag is Material material)

{

ManufacturerPage page = new ManufacturerPage(material);

Manager.MainFrame.Navigate(page);

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

try

{

Calculation calculation = new Calculation();

Result.Text = (Convert.ToInt32(TBoxSquare.Text) / Convert.ToInt32(\_curentMaterial.Square) \* Convert.ToInt32(\_curentMaterial.Price)).ToString() + " рублей";

calculation.Price = Convert.ToInt32(TBoxSquare.Text) / Convert.ToInt32(\_curentMaterial.Square) \* Convert.ToInt32(\_curentMaterial.Price);

calculation.UserID = Manager.CurrentUser.UserID;

calculation.MaterialID = \_curentMaterial.MaterialID;

using (var transaction =

MaterialsCalculatorEntities.GetContext().Database.BeginTransaction())

{

try

{

MaterialsCalculatorEntities.GetContext().Calculations.Add(calculation);

MaterialsCalculatorEntities.GetContext().SaveChanges();

transaction.Commit();

MessageBox.Show("Запись добавленна");

}

catch (Exception ex)

{

transaction.Rollback();

MessageBox.Show(ex.Message.ToString());

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

if (sender is Button button && button.Tag is Material material)

{

\_curentMaterial = material;

}

}

}

}

Окно типов материалов

<Page x:Class="MaterialsCalculator.Pages.MaterialTypesPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Мероприятия"

IsVisibleChanged="Page\_IsVisibleChanged"

Style="{StaticResource base\_page}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch"

Background="{StaticResource additional}">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Введите название типа материала:"

Style="{StaticResource base\_textblock}" />

<TextBox Width="300" x:Name="TBoxSearch"

TextChanged="TBoxSearch\_TextChanged"/>

</StackPanel>

</WrapPanel>

<TextBlock Grid.Row="2" x:Name="TextBlockInfo"

Width="300" HorizontalAlignment="Left" />

<ListBox x:Name="ListBoxMaterialTypes"

ScrollViewer.VerticalScrollBarVisibility="Visible"

SelectedValuePath="AgentId"

Grid.Row="1" Grid.Column="0">

<ListBox.ItemTemplate>

<DataTemplate >

<Grid HorizontalAlignment="Stretch" UseLayoutRounding="True">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="550"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Image Source="{Binding Path=GetPicture}" Grid.Row="0" Stretch="UniformToFill" HorizontalAlignment="Left" Grid.RowSpan="2">

</Image>

<TextBlock

Text="{Binding Path=Name}"

VerticalAlignment="center" TextAlignment="Center" Width="100" TextWrapping="Wrap" HorizontalAlignment="Left" FontSize="26" Grid.Row="0" Grid.Column="1" />

<TextBlock

Text="{Binding Path=Description}"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap" HorizontalAlignment="Left" FontSize="14" Grid.Row="2" Grid.Column="1"/>

<Button Content="{Binding Path=Name}" Click="MaterialTypeButton\_Click" Grid.Column="2" HorizontalAlignment="Right" Width="100">

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

</Grid>

</Page>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace MaterialsCalculator.Pages

{

/// <summary>

/// Логика взаимодействия для MaterialTypesPage.xaml

/// </summary>

public partial class MaterialTypesPage : Page

{

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//обновление данных после каждой активации окна

if (Visibility == Visibility.Visible)

{

MaterialsCalculatorEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

}

}

private void TBoxSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

int \_itemcount = 0; // количество записей в таблице Events

public MaterialTypesPage()

{

InitializeComponent();

LoadAndInitData();

}

/// <summary>

/// Загрузка и подготовка данных страницы

/// </summary>

void LoadAndInitData()

{

// загрузка мероприятий в ListBox сортируем по Дате

var mce = MaterialsCalculatorEntities.GetContext().MaterialTypes.ToList();

ListBoxMaterialTypes.ItemsSource = mce;

// запоминаем общще количество мероприятий в БД

\_itemcount = ListBoxMaterialTypes.Items.Count;

TextBlockInfo.Text = $" Результат запроса: {\_itemcount} записей из {\_itemcount}";

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем актуальный список мероприятий из БД

var currentEvents = MaterialsCalculatorEntities.GetContext().MaterialTypes.OrderBy(p => p.MaterialTypeID).ToList();

// выбор тех мероприятий, в названии которых есть поисковая строка

currentEvents = currentEvents.Where(p => p.Name.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// В качестве источника данных присваиваем список данных

ListBoxMaterialTypes.ItemsSource = currentEvents;

// отображение количества записей

TextBlockInfo.Text = $" Результат запроса: {currentEvents.Count} " +

$"записей из {\_itemcount}";

}

private void MaterialTypeButton\_Click(object sender, RoutedEventArgs e)

{

if (sender is Button button && button.Tag is MaterialType materialType)

{

// Открываете новую страницу с выбранным типом материала

OpenNewPageWithMaterialType(materialType);

}

}

private void OpenNewPageWithMaterialType(MaterialType materialType)

{

// Создаете новую страницу с выбранным типом материала

MaterialPage detailsPage = new MaterialPage(materialType);

// Навигируете к новой странице

if (Manager.MainFrame != null)

{

Manager.MainFrame.Navigate(detailsPage);

}

}

}

}

Окно юзера

<Page x:Class="MaterialsCalculator.Pages.UsersPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Style="{StaticResource base\_page}"

Title="Окно юзера" Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="14.906"/>

<RowDefinition Height="60.094"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}"/>

<Image Margin="10,10,0,10"

HorizontalAlignment="Left"/>

<TextBlock Text="Окно юзера"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20"

FontWeight="Bold"/>

<StackPanel Margin="10,10,10,10" Grid.Row="1" Grid.RowSpan="2">

<TextBlock x:Name="TBLastName" Text="{Binding User.LastName}" FontSize="18" FontWeight="Bold" Margin="0,0,0,10"/>

<TextBlock x:Name="TBFirstName" Text="{Binding User.FirstName}" Margin="0,0,0,5"/>

<TextBlock x:Name="TBMiddleName" Text="{Binding User.MiddleName}" Margin="0,0,0,5"/>

</StackPanel>

<ListBox x:Name="ListBoxMaterials"

ScrollViewer.VerticalScrollBarVisibility="Visible"

SelectedValuePath="AgentId"

Grid.Row="3" Grid.Column="0" >

<ListBox.ItemTemplate>

<DataTemplate >

<Grid HorizontalAlignment="Stretch" UseLayoutRounding="True">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >

<RowDefinition/>

</Grid.RowDefinitions>

<Image Source="{Binding Path=Material.GetImage}" Grid.Row="0" Stretch="UniformToFill" HorizontalAlignment="Left" Grid.RowSpan="2">

</Image>

<TextBlock

Text="{Binding Path=Material.Name}"

VerticalAlignment="center" TextAlignment="Center" Width="100" TextWrapping="Wrap"

HorizontalAlignment="Center" FontSize="20" Grid.Row="0" Grid.Column="1" />

<TextBlock

Text="{Binding Path=Price}"

VerticalAlignment="center" TextAlignment="Center" Width="100" TextWrapping="Wrap"

HorizontalAlignment="Center" FontSize="20" Grid.Row="0" Grid.Column="2"/>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<Grid Grid.Row="4" Background="{StaticResource additional}">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="83\*"/>

<ColumnDefinition Width="77\*"/>

</Grid.ColumnDefinitions>

</Grid>

</Grid>

</Page>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace MaterialsCalculator.Pages

{

/// <summary>

/// Логика взаимодействия для UsersPage.xaml

/// </summary>

public partial class UsersPage : Page

{

public User user;

public UsersPage(User user)

{

InitializeComponent();

this.user = user;

this.LoadData();

}

private void LoadData()

{

TBLastName.Text = $"{user.LastName}";

TBFirstName.Text = $"{user.FirstName}";

TBMiddleName.Text = $"{user.MiddleName}";

// загрузка мероприятий в ListBox сортируем по Дате

var mce = MaterialsCalculatorEntities.GetContext().Calculations.ToList().Where(p => p.User.UserID == user.UserID);

ListBoxMaterials.ItemsSource = mce;

}

}

}

Окно добавления материала

<Window x:Class="MaterialsCalculator.Windows.AddMaterialWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MaterialsCalculator.Windows"

mc:Ignorable="d"

Title="AddMaterialWindow" Height="450" Width="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="120"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}" Grid.ColumnSpan="4"/>

<TextBlock Grid.Row="0" Text="Добавить материал"

Grid.ColumnSpan="4"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20"

FontWeight="Bold"/>

<TextBlock Grid.Row="1" Text="Название:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBName" Grid.Column="1" Grid.Row="1"/>

<TextBlock Grid.Row="2" Text="Описание:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBDescription" Grid.Column="1" Grid.Row="2"/>

<TextBlock Grid.Row="3" Text="Фото:"

Style="{StaticResource add\_window\_textblock}"/>

<Image x:Name="ImageMaterialPhoto" Grid.Row="8" Grid.Column="1" Grid.RowSpan="7" />

<Button x:Name="ButtonLoadPhoto" Grid.Column="2" Grid.Row="3" Width="200"

HorizontalAlignment="Left" Margin="0"

Content="Загрузить фото" Click="ButtonLoadPhoto\_Click"/>

<TextBlock Grid.Row="4" Text="Тип матерьяла:"

Style="{StaticResource add\_window\_textblock}"/>

<ComboBox x:Name="ComboBoxType" Grid.Row="4" Grid.Column="1" >

<ComboBoxItem Content="Tile"/>

<ComboBoxItem Content="Roof"/>

<ComboBoxItem Content="Logs"/>

</ComboBox>

<TextBlock Grid.Row="5" Text="Производитель:"

Style="{StaticResource add\_window\_textblock}"/>

<ComboBox x:Name="ComboBoxManufacturer" Grid.Row="5" Grid.Column="1" >

<ComboBoxItem Content="TileCrafter"/>

<ComboBoxItem Content="RoofWorks"/>

<ComboBoxItem Content="LogsMasters"/>

</ComboBox>

<TextBlock Grid.Row="6" Text="Площадь:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBSquare" Grid.Column="1" Grid.Row="6"/>

<TextBlock Grid.Row="7" Grid.Column="0" Text="Цена:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBPrice" Grid.Column="1" Grid.Row="7" />

<StackPanel Orientation="Horizontal"

Grid.Row="13" Grid.Column="3"

HorizontalAlignment="Center">

<Button Width="100" Height="30"

Content="OK"

x:Name="BtnOk" Click="BtnOk\_Click"/>

<Button Width="100"

Height="30"

Content="Отмена"

x:Name="BtnCancel" Click="BtnCancel\_Click" />

</StackPanel>

</Grid>

</Window>

using MaterialsCalculator.Models;

using Microsoft.Win32;

using System;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System.Windows.Media.Imaging;

namespace MaterialsCalculator.Windows

{

/// <summary>

/// Логика взаимодействия для AddMaterialWindow.xaml

/// </summary>

public partial class AddMaterialWindow : Window

{

private Material \_currentMaterial = new Material();

// путь к загружаемой фтографии

private string \_filePath = null;

// имя файла

private string \_photoName = null;

// путь к папке с фотографиями модераторов

public AddMaterialWindow()

{

InitializeComponent();

this.DataContext = \_currentMaterial;

}

/\*void LoadAndInitData()

{

// подгружаем данные выпадающих списков

// страны

ComboBoxType.ItemsSource = MaterialsCalculatorEntities.GetContext().MaterialTypes.ToList();

}\*/

/// <summary>

/// Проверка полей ввод на корректыне данные

/// </summary>

/// <returns>текст StringBuilder содержащий ошибки, если они есть</returns>

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

using (var transaction =

MaterialsCalculatorEntities.GetContext().Database.BeginTransaction())

{

try

{

\_currentMaterial.Name = TBName.Text;

\_currentMaterial.Description = TBDescription.Text;

\_currentMaterial.MaterialTypeID = ComboBoxType.SelectedIndex+1;

\_currentMaterial.ManufacturerID = ComboBoxManufacturer.SelectedIndex+1;

\_currentMaterial.Square = Convert.ToInt32(TBSquare.Text).ToString();

\_currentMaterial.Price = Convert.ToInt32(TBPrice.Text).ToString();

string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\Materials\";

string photo = ChangePhotoName(\_currentDirectory);

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentMaterial.Image = photo;

MaterialsCalculatorEntities.GetContext().Materials.Add(\_currentMaterial);

MaterialsCalculatorEntities.GetContext().SaveChanges();

transaction.Commit();

// Сохраняем изменения в БД

MessageBox.Show("Запись добавлена");

DialogResult = true; // Возвращаемся на предыдущую форму

}

catch (Exception ex)

{

transaction.Rollback();

MessageBox.Show(ex.Message.ToString());

}

}

}

//подбор имени файла

string ChangePhotoName(string \_currentDirectory)

{

string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{

while (File.Exists(x))

{

i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

DialogResult = false;

}

private void ButtonLoadPhoto\_Click(object sender, RoutedEventArgs e)

{

try

{

//Диалог открытия файла

OpenFileDialog op = new OpenFileDialog

{

Title = "Select a picture",

Filter = "PNG Files (\*.png)|\*.png"

};

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{

ImageMaterialPhoto.Source = new BitmapImage(new Uri(op.FileName));

\_photoName = op.SafeFileName;

\_filePath = op.FileName;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK,

MessageBoxImage.Error);

\_filePath = null;

}

}

}

}

Окно регистрации

<Window x:Class="MaterialsCalculator.Windows.AutorizationWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MaterialsCalculator.Windows"

mc:Ignorable="d"

Title="Вход" Style="{StaticResource base\_window}" Height="350" Width="400" MinHeight="200" MinWidth="400"

WindowStartupLocation="CenterScreen" ResizeMode="NoResize" >

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="34"/>

<RowDefinition Height="34"/>

<!--,Будем показывать и скрывать капчу, при неудачном входе в систему-->

<RowDefinition x:Name="RowCaptcha1" Height="34"/>

<RowDefinition x:Name="RowCaptcha2" />

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}" />

<Image Margin="10" HorizontalAlignment="Left" />

<TextBlock Grid.Row="0" Text="Авторизация"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

Background="{StaticResource additional}"

FontSize="20"

FontWeight="Bold" />

<StackPanel Orientation="Horizontal" Grid.Row="1">

<TextBlock Style="{StaticResource item\_textblock}"

Grid.Row="1"

Text="Email"

Margin="20 0"

VerticalAlignment="Center" Width="120"/>

<TextBox x:Name="TbLogin"

Grid.Column="1"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Row="2">

<TextBlock Style="{StaticResource item\_textblock}"

Grid.Row="2" Grid.Column="0"

Text="Пароль" Margin="20 0"

VerticalAlignment="Center" Width="120"/>

<PasswordBox x:Name="TbPass" Height="30" Width="200" />

</StackPanel >

<StackPanel Orientation="Horizontal" Grid.Row="3">

<TextBlock Style="{StaticResource item\_textblock}"

Grid.Row="2" Grid.Column="0"

Text="Введите капчу" Margin="20 0"

VerticalAlignment="Center" Width="120"/>

<TextBox x:Name="TbCaptcha"

Grid.Column="1"

Width="200"/>

</StackPanel >

<Image Grid.Row="4" x:Name="ImageCaptcha" Height="100"

Width="300"

VerticalAlignment="Top"/>

<Button Width="24" Height="24" Padding="0"

VerticalContentAlignment="Top"

Content="O" HorizontalAlignment="Left"

VerticalAlignment="Top"

x:Name="BtnRenewCaptcha"

Click="BtnRenewCaptcha\_Click" Grid.Row="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="5" HorizontalAlignment="Center">

<Button Width="100" Height="30"

Content="Ок"

x:Name="BtnOk"

Click="BtnOkClick"/>

<Button Width="100"

Height="30"

Content="Закрыть"

x:Name="BtnCancel"

Click="BtnCancelClick"/>

<Button Width="100" Height="30"

Content="Регистрация"

x:Name="BtnRegistration"

Click="BtnRegistration\_Click"/>

</StackPanel>

<TextBlock Style="{StaticResource item\_textblock}"

Grid.Row="4"

x:Name="TextBlockTime"

HorizontalAlignment="Center"

VerticalAlignment="Bottom" Width="326"/>

</Grid>

</Window>

using MaterialsCalculator.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Threading;

namespace MaterialsCalculator.Windows

{

/// <summary>

/// Логика взаимодействия для AutorizationWindow.xaml

/// </summary>

public partial class AutorizationWindow : Window

{

// Переменная флаг, меняет свое значение,

// если пользователь не смог ввести с первого раза пароль и логин

// если b == true, то отобразим капчу

bool b = false;

int count = 0;

// таймер

DispatcherTimer timer = new DispatcherTimer();

int seconds = 0; // секунды

string captcha = ""; // текст капчи

public AutorizationWindow()

{

InitializeComponent();

LoadAndInitData();

}

/// <summary>

/// Загружает и инициализирует данные

/// </summary>

void LoadAndInitData()

{

this.Height = 200; // задаем высоту окна

timer.Tick += timer\_Tick; // к событию Tick таймера привязываем обработчик события

// скрываем строки с капчой

RowCaptcha1.Height = new GridLength(0);

RowCaptcha2.Height = new GridLength(0);

//TbLogin.Text = "loginDEpxl2018";

//TbPass.Password = "P6h4Jq";

}

// обаботчик события срабатывает через каждые т секунд

void timer\_Tick(object sender, EventArgs e)

{

seconds -= 1;

TextBlockTime.Text = $"Осталось {seconds} секунд до разблокировки";

if (seconds == 0) // оставливаем таймер, разблокировываем кнопку

{

BtnOk.IsEnabled = true;

TextBlockTime.Text = "";

timer.Stop();

}

}

private void BtnOkClick(object sender, RoutedEventArgs e)

{

try

{ //загрузка всех пользователей из БД в список

List<User> users = MaterialsCalculatorEntities.GetContext().Users.ToList();

User user = users.FirstOrDefault(p => p.Password == TbPass.Password && p.Email.ToString() == TbLogin.Text);

//User user = users.FirstOrDefault(p => p.Password == "User!1" && p.Email.ToString() == "user@user.com");

//User user = users.FirstOrDefault(p => p.Password == "Admin!1" && p.Email.ToString() == "admin@admin.com");

// удачный вход после ввода пароля и логина или пароля, логина и капчи

if ((user != null && !b) || (user != null && b && TbCaptcha.Text == captcha))

{

Manager.CurrentUser = user;

// логин и пароль корректные запускаем главную форму приложения

DialogResult = true;

}

else

{

MessageBox.Show("Не верный логин или пароль");

count++;

if (count >= 3)

{

// меняем высоту формы

this.Height = 350;

// вызываем метод отображения капчи

ShowCaptcha();

// задаем параметры таймера, событие Tick срабатывает через каждую секунду

timer.Interval = TimeSpan.FromSeconds(1);

// блокируем кнопку

BtnOk.IsEnabled = false;

// на 10 секунд

seconds = 10;

// отображает сколько секунд осталось до разблокировки

TextBlockTime.Text = $"Осталось {seconds} секунд до разблокировки";

// включаем таймер

timer.Start();

RowCaptcha1.Height = new GridLength(34);

RowCaptcha2.Height = new GridLength(1, GridUnitType.Star);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

//код кнопки Cancel

private void BtnCancelClick(object sender, RoutedEventArgs e)

{

DialogResult = false;

this.Close();

}

// попытка закрыть приложение

private void BtnRenewCaptcha\_Click(object sender, RoutedEventArgs e)

{

// кнопка обновить капчу

ShowCaptcha();

}

/// <summary>

/// отображает капчу

/// </summary>

void ShowCaptcha()

{

// из класса MakeCaptcha вызываем метод CreateImage

var tuple = MakeCaptcha.CreateImage(300, 100, 4);

// получаем ImageSource

ImageCaptcha.Source = tuple.image;

// получаем текст капчи

captcha = tuple.captcha;

}

private void BtnRegistration\_Click(object sender, RoutedEventArgs e)

{

RegistrationWindow regWindow = new RegistrationWindow();

this.Close();

regWindow.ShowDialog();

}

}

}

Окно удаления материала

<Window x:Class="MaterialsCalculator.Windows.DeleteMaterialsWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MaterialsCalculator.Windows"

mc:Ignorable="d"

Title="{Binding ElementName=MainFrame, Path=Content.Title}"

WindowStartupLocation="CenterScreen"

Style="{StaticResource base\_window}"

Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}"/>

<Image Margin="10" HorizontalAlignment="Left"/>

<TextBlock Grid.Row="0" Text="Удаление материала"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20" FontWeight="Bold"/>

<ListBox x:Name="ListBoxMaterials"

ScrollViewer.VerticalScrollBarVisibility="Visible"

SelectedValuePath="AgentId"

Grid.Row="1" Grid.Column="0" Grid.RowSpan="4">

<ListBox.ItemTemplate>

<DataTemplate >

<Grid HorizontalAlignment="Stretch" UseLayoutRounding="True">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="100"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Image Source="{Binding Path=GetImage}" Grid.Row="0" Stretch="UniformToFill" HorizontalAlignment="Left" Grid.RowSpan="2">

</Image>

<TextBlock

Text="{Binding Path=Name}"

VerticalAlignment="center" TextAlignment="Center" Width="100" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="26" Grid.Row="0" Grid.Column="1" />

<TextBlock

Text="{Binding Path=Description}"

VerticalAlignment="center" TextAlignment="Center" Width="auto" TextWrapping="Wrap"

HorizontalAlignment="Left" FontSize="14" Grid.Row="3" Grid.Column="1"/>

<Button Content="Удалить" Click="Button\_Click" Grid.Column="2" Grid.Row="2"

HorizontalAlignment="Right" Width="100">

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

<Button Content="Изменить" Click="Button\_Click\_1" Grid.Column="2" Grid.Row="3"

HorizontalAlignment="Right" Width="100">

<Button.Tag>

<Binding RelativeSource="{RelativeSource Mode=Self}" Path="DataContext" />

</Button.Tag>

</Button>

</Grid>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<TextBlock x:Name="TextBlockName" Width="auto"

Height="40" HorizontalAlignment="Right"

VerticalAlignment="Top"/>

<Grid Grid.Row="2" Background="{StaticResource additional}"/>

</Grid>

</Window>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System;

namespace MaterialsCalculator.Windows

{

/// <summary>

/// Логика взаимодействия для DeleteMaterialsWindow.xaml

/// </summary>

public partial class DeleteMaterialsWindow : Window

{

public Material \_curentMaterial;

public DeleteMaterialsWindow()

{

InitializeComponent();

LoadAndInitData();

}

void LoadAndInitData()

{

// загрузка мероприятий в ListBox сортируем по Дате

var mce = MaterialsCalculatorEntities.GetContext().Materials.ToList();

ListBoxMaterials.ItemsSource = mce;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

// Проверяем, выбран ли какой-то материал для удаления

if (ListBoxMaterials.SelectedItem != null)

{

// Получаем выбранный материал

Material material = ListBoxMaterials.SelectedItem as Material;

// Отображаем диалоговое окно для подтверждения удаления

MessageBoxResult result = MessageBox.Show("Вы уверены, что хотите удалить выбранный материал?", "Подтверждение удаления", MessageBoxButton.YesNo, MessageBoxImage.Question);

// Если пользователь подтвердил удаление

if (result == MessageBoxResult.Yes)

{

// Удаляем материал из базы данных

using (var context = new MaterialsCalculatorEntities())

{

// При условии, что MaterialId - это уникальный идентификатор материала в вашей базе данных

var materialToDelete = context.Materials.FirstOrDefault(m => m.MaterialID == material.MaterialID);

if (materialToDelete != null)

{

context.Materials.Remove(materialToDelete);

context.SaveChanges();

// Обновляем отображаемый список материалов после удаления

LoadAndInitData();

MessageBox.Show("Материал успешно удален", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("Материал не найден", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

else

{

MessageBox.Show("Выберите материал для удаления", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

//событие попытки закрытия окна,

// если пользователь выберет Cancel, то форму не закроем

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

// Проверяем, выбран ли какой-то материал для редактирования

if (ListBoxMaterials.SelectedItem != null)

{

// Получаем выбранный материал

Material material = ListBoxMaterials.SelectedItem as Material;

// Создаем окно редактирования и передаем выбранный материал

EditMaterialWindow editWindow = new EditMaterialWindow(material);

editWindow.ShowDialog();

}

else

{

MessageBox.Show("Выберите материал для редактирования", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

}

Окно регистрации материала

<Window x:Class="MaterialsCalculator.Windows.EditMaterialWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MaterialsCalculator.Windows"

mc:Ignorable="d"

Title="EditMaterialWindow" Height="450" Width="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="120"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}" Grid.ColumnSpan="4"/>

<TextBlock Grid.Row="0" Text="Редактировать матерьял"

Grid.ColumnSpan="4"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20"

FontWeight="Bold"/>

<TextBlock Grid.Row="1" Text="Название:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBName" Grid.Column="1" Grid.Row="1"/>

<TextBlock Grid.Row="2" Text="Описание:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBDescription" Grid.Column="1" Grid.Row="2"/>

<TextBlock Grid.Row="3" Text="Фото:"

Style="{StaticResource add\_window\_textblock}"/>

<Image x:Name="ImageMaterialPhoto" Grid.Row="8" Grid.Column="1" Grid.RowSpan="7" />

<Button x:Name="ButtonLoadPhoto" Grid.Column="2" Grid.Row="3" Width="200"

HorizontalAlignment="Left" Margin="0"

Content="Загрузить фото" Click="ButtonLoadPhoto\_Click"/>

<TextBlock Grid.Row="4" Text="Тип матерьяла:"

Style="{StaticResource add\_window\_textblock}"/>

<ComboBox x:Name="ComboBoxType" Grid.Row="4" Grid.Column="1" >

<ComboBoxItem Content="Tile"/>

<ComboBoxItem Content="Roof"/>

<ComboBoxItem Content="Logs"/>

</ComboBox>

<TextBlock Grid.Row="5" Text="Производитель:"

Style="{StaticResource add\_window\_textblock}"/>

<ComboBox x:Name="ComboBoxManufacturer" Grid.Row="5" Grid.Column="1" >

<ComboBoxItem Content="TileCrafter"/>

<ComboBoxItem Content="RoofWorks"/>

<ComboBoxItem Content="LogsMasters"/>

</ComboBox>

<TextBlock Grid.Row="6" Text="Площадь:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBSquare" Grid.Column="1" Grid.Row="6"/>

<TextBlock Grid.Row="7" Grid.Column="0" Text="Цена:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TBPrice" Grid.Column="1" Grid.Row="7" />

<StackPanel Orientation="Horizontal"

Grid.Row="13" Grid.Column="3"

HorizontalAlignment="Center">

<Button Width="100" Height="30"

Content="OK"

x:Name="BtnOk" Click="BtnOk\_Click"/>

<Button Width="100"

Height="30"

Content="Отмена"

x:Name="BtnCancel" Click="BtnCancel\_Click" />

</StackPanel>

</Grid>

</Window>

using MaterialsCalculator.Models;

using Microsoft.Win32;

using System;

using System.IO;

using System.Windows;

using System.Windows.Media.Imaging;

namespace MaterialsCalculator.Windows

{

public partial class EditMaterialWindow : Window

{

private Material \_currentMaterial;

private string \_filePath = null;

private string \_photoName = null;

public EditMaterialWindow(Material material)

{

InitializeComponent();

//LoadData();

\_currentMaterial = material;

this.DataContext = \_currentMaterial;

// Заполнение данных о материале в поля формы

TBName.Text = \_currentMaterial.Name;

TBDescription.Text = \_currentMaterial.Description;

ComboBoxType.SelectedIndex = (int)\_currentMaterial.MaterialTypeID-1;

ComboBoxManufacturer.SelectedIndex = (int)\_currentMaterial.ManufacturerID-1;

TBPrice.Text = \_currentMaterial.Price;

TBSquare.Text = \_currentMaterial.Square;

ImageMaterialPhoto.Source = new BitmapImage(new Uri(Directory.GetCurrentDirectory() + @"\Images\Materials\" +\_currentMaterial.Image));

}

public void LoadData()

{

/\*\_currentMaterial = material;

this.DataContext = \_currentMaterial;

// Заполнение данных о материале в поля формы

TBName.Text = \_currentMaterial.Name;

TBDescription.Text = \_currentMaterial.Description;

// Другие поля...

// Отображение изображения материала, если оно есть

if (!string.IsNullOrEmpty(\_currentMaterial.Image))

{

ImageMaterialPhoto.Source = new BitmapImage(new Uri(\_currentMaterial.Image));

}\*/

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

try

{

// Обновление данных о материале

\_currentMaterial.Name = TBName.Text;

\_currentMaterial.Description = TBDescription.Text;

\_currentMaterial.MaterialTypeID = ComboBoxType.SelectedIndex + 1;

\_currentMaterial.ManufacturerID = ComboBoxManufacturer.SelectedIndex + 1;

\_currentMaterial.Square = Convert.ToInt32(TBSquare.Text).ToString();

\_currentMaterial.Price = Convert.ToInt32(TBPrice.Text).ToString();

string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\Materials\";

string photo = ChangePhotoName(\_currentDirectory);

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentMaterial.Image = photo;

// Сохранение изменений в базе данных

MaterialsCalculatorEntities.GetContext().SaveChanges();

MessageBox.Show("Изменения сохранены");

DialogResult = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

DialogResult = false;

}

string ChangePhotoName(string \_currentDirectory)

{

string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{

while (File.Exists(x))

{

i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

private void ButtonLoadPhoto\_Click(object sender, RoutedEventArgs e)

{

try

{

//Диалог открытия файла

OpenFileDialog op = new OpenFileDialog

{

Title = "Select a picture",

Filter = "PNG Files (\*.png)|\*.png"

};

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{

ImageMaterialPhoto.Source = new BitmapImage(new Uri(op.FileName));

\_photoName = op.SafeFileName;

\_filePath = op.FileName;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK,

MessageBoxImage.Error);

\_filePath = null;

}

}

}

}

Основная страница

<Window x:Class="MaterialsCalculator.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MaterialsCalculator"

mc:Ignorable="d"

Title="{Binding ElementName=MainFrame, Path=Content.Title}"

WindowStartupLocation="CenterScreen"

Style="{StaticResource base\_window}"

Height="450" Width="800"

Closing="WindowClosing" Closed="WindowClosed" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}"/>

<Image Margin="10" HorizontalAlignment="Left"/>

<TextBlock Grid.Row="0" Text="Калькулятор строительных материалов"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20" FontWeight="Bold"/>

<Frame x:Name="MainFrame" Grid.Row="1"

BorderBrush="{StaticResource akcent}"

NavigationUIVisibility="Hidden"

BorderThickness="1" />

<Button x:Name="BtnEnter"

Content="Авторизация" Grid.Row="2"

HorizontalAlignment="Left" Margin="150,10,0,10" Width="100" Panel.ZIndex="100"

Click="BtnEnter\_Click" />

<Button x:Name="BtnHome"

Content="Домой" Grid.Row="2"

HorizontalAlignment="Center" Margin="0,10,0,10" Width="100" Panel.ZIndex="100"

Click="BtnHome\_Click" />

<Button x:Name="BtnUser"

Content="Юзер" Grid.Row="2"

HorizontalAlignment="Right" Margin="0,10,150,10" Width="100" Panel.ZIndex="100"

Click="BtnUser\_Click" />

<TextBlock x:Name="TextBlockName" Width="auto"

Height="40" HorizontalAlignment="Right"

VerticalAlignment="Top"/>

<Grid Grid.Row="2" Background="{StaticResource additional}"/>

</Grid>

</Window>

using MaterialsCalculator.Models;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

using System;

using System.Windows;

namespace MaterialsCalculator

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

User \_currentUser; //текущий пользователь в системе

public MainWindow()

{

InitializeComponent();

LoadData();

}

/// <summary>

/// загружаем данные и инициализируем переменные

/// </summary>

void LoadData()

{

MainFrame.Navigate(new MaterialTypesPage());

Manager.MainFrame = MainFrame;

}

/\*void UpdateData(Object o)

{

Manager.MainFrame = o;

}\*/

private void WindowClosed(object sender, EventArgs e)

{

// показать владельца формы

//Owner.Show();

// или если надо, то можно закрыть приложение командой

Application.Current.Shutdown();

}

//событие попытки закрытия окна,

// если пользователь выберет Cancel, то форму не закроем

private void WindowClosing(object sender, System.ComponentModel.CancelEventArgs e)

{

MessageBoxResult x = MessageBox.Show("Вы действительно хотите выйти?", "Выйти",

MessageBoxButton.OKCancel, MessageBoxImage.Question);

if (x == MessageBoxResult.Cancel)

e.Cancel = true;

}

//авторизация

private void BtnEnter\_Click(object sender, RoutedEventArgs e)

{

try

{

AutorizationWindow window = new AutorizationWindow();

// открытие формы в режиме модального окна

if (window.ShowDialog() == true)

{

// сюда попадем, если пользователь введёт корректные учётные данные

// получаем RoleId пользователя

\_currentUser = Manager.CurrentUser;

int role = Convert.ToInt32(\_currentUser.RoleID);

if (role == 2) //если 1, то участник и открываем окно участника

{

UsersPage usersPage = new UsersPage(\_currentUser);

Manager.MainFrame.Navigate(usersPage);

}

if (role == 1) //если 2, то модератор и открываем окно модератора

{

AdminPage adminPage = new AdminPage(\_currentUser);

Manager.MainFrame.Navigate(adminPage);

}

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void BtnHome\_Click(object sender, RoutedEventArgs e)

{

this.LoadData();

}

private void BtnUser\_Click(object sender, RoutedEventArgs e)

{

try

{

int role = Convert.ToInt32(\_currentUser.RoleID);

if (role == 2) //если 1, то участник и открываем окно участника

{

UsersPage usersPage = new UsersPage(\_currentUser);

Manager.MainFrame.Navigate(usersPage);

}

if (role == 1) //если 2, то модератор и открываем окно модератора

{

AdminPage adminPage = new AdminPage(\_currentUser);

Manager.MainFrame.Navigate(adminPage);

}

}

catch (Exception ex)

{

MessageBox.Show("Авторизируйтесь, пожалуйста");

}

}

}

}

Окно регистрации

<Window x:Class="MaterialsCalculator.Windows.RegistrationWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MaterialsCalculator.Windows"

xmlns:xctk="https://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

Title="Регистрация" WindowStartupLocation="CenterScreen"

Height="500" Width="800"

MinHeight="500" MinWidth="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="120"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="60"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid Background="{StaticResource additional}" Grid.ColumnSpan="4"/>

<TextBlock Grid.Row="0" Text="Регистрация"

Grid.ColumnSpan="4"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="{StaticResource main}"

FontSize="20"

FontWeight="Bold"/>

<TextBlock Grid.Row="1" Text="ФИО:"

Style="{StaticResource add\_window\_textblock}"/>

<StackPanel Grid.Row="1" Grid.Column="1"

Orientation="Horizontal"

HorizontalAlignment="Stretch" >

<TextBox x:Name="TextBoxLastName" Width="100"

Text="{Binding Surname}"/>

<TextBox x:Name="TextBoxFirstName" Width="100"

Text="{Binding Name}"/>

<TextBox x:Name="TextBoxMiddleName" Width="100"

Text="{Binding Patronymic}"/>

</StackPanel>

<TextBlock Grid.Row="2" Text="Пол:"

Style="{StaticResource add\_window\_textblock}"/>

<ComboBox x:Name="ComboBoxGender" Grid.Row="2" Grid.Column="1" >

<ComboBoxItem Content="женский"/>

<ComboBoxItem Content="мужской"/>

</ComboBox>

<TextBlock Grid.Row="3" Text="Email:"

Style="{StaticResource add\_window\_textblock}"/>

<TextBox x:Name="TextBoxEmail" Grid.Row="3" Grid.Column="1"

Text="{Binding Email}"/>

<TextBlock Grid.Row="4" Grid.Column="0" Text="Пароль:"

Style="{StaticResource add\_window\_textblock}"/>

<PasswordBox x:Name="PasswordBoxFirst" Grid.Row="4" Grid.Column="1" />

<TextBox Name="TextBoxPasswordFirst"

Visibility="Collapsed" Grid.Row="4" Grid.Column="1" />

<TextBlock Grid.Row="5" Grid.Column="0" Text="Повтор пароль:"

Style="{StaticResource add\_window\_textblock}"/>

<PasswordBox x:Name="PasswordBoxAgain" Grid.Row="5" Grid.Column="1" />

<TextBox Name="TextBoxPasswordAgain" Visibility="Collapsed"

Grid.Row="6" Grid.Column="1" />

<!--<CheckBox x:Name="CheckBoxIsVisiblePassword"

Grid.Row="7" Grid.Column="1"

VerticalAlignment="Top"

Margin="10 0 0 0" Content="Видимый пароль"

Checked="CheckBoxIsVisiblePassword\_Checked"

Unchecked="CheckBoxIsVisiblePassword\_Unchecked" />-->

<StackPanel Orientation="Horizontal"

Grid.Row="13" Grid.Column="3"

HorizontalAlignment="Center">

<Button Width="100" Height="30"

Content="OK"

x:Name="BtnOk" Click="BtnOk\_Click"/>

<Button Width="100"

Height="30"

Content="Отмена"

x:Name="BtnCancel" Click="BtnCancel\_Click" />

</StackPanel>

</Grid>

</Window>

using MaterialsCalculator.Models;

using Microsoft.Win32;

using System;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using MaterialsCalculator.Pages;

using MaterialsCalculator.Windows;

namespace MaterialsCalculator.Windows

{

/// <summary>

/// Логика взаимодействия для RegistrationWindow.xaml

/// </summary>

public partial class RegistrationWindow : Window

{

// создаваемый новый пользователь

private User \_currentUser = new User();

public RegistrationWindow()

{

InitializeComponent();

LoadAndInitData();

}

// загрузка и задание начальных значений полям

void LoadAndInitData()

{

// формируем новый IdNumber нового пользоватля

// получам количество пользователей

int k = MaterialsCalculatorEntities.GetContext().Users.Count();

int numberid = 1;

if (k > 0)

numberid = Convert.ToInt32(MaterialsCalculatorEntities.GetContext().Users.

Max(x => x.UserID)) + 1;

this.DataContext = \_currentUser;

}

bool CheckPassword(string password)

{

// проверка длины пароля

if (password.Length < 6)

return false;

// проверка заглавных и строчных букв

if (password.ToLower() == password || password.ToUpper() == password)

return false;

int count = 0;

int spec = 0;

for (int i = 0; i < password.Length; i++)

{

// подсчёт цифр

if (char.IsDigit(password[i]))

count++;

// подсчёт спецсимволов

if (!char.IsLetterOrDigit(password[i]))

spec++;

}

//MessageBox.Show($"цифр {count} спец {spec}");

// проверка не менее одного спецсимвола и не менее одной цифры

if (count == 0 || spec == 0)

return false;

return true;

}

/// <summary>

/// Проверка полей ввод на корректыне данные

/// </summary>

/// <returns>текст StringBuilder содержащий ошибки, если они есть</returns>

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

// проверка полей на содержимое

if (string.IsNullOrWhiteSpace(TextBoxLastName.Text))

s.AppendLine("Укажите Фамилию");

if (string.IsNullOrWhiteSpace(TextBoxFirstName.Text))

s.AppendLine("Укажите Имя");

if (ComboBoxGender.SelectedIndex == -1)

s.AppendLine("Выберите пол");

if (string.IsNullOrWhiteSpace(TextBoxEmail.Text))

s.AppendLine("Укажите Email");

// проверка пароля

if (string.IsNullOrWhiteSpace(PasswordBoxFirst.Password) &&

string.IsNullOrWhiteSpace(TextBoxPasswordFirst.Text))

s.AppendLine("Укажите пароль");

else

{

if (PasswordBoxFirst.Password != PasswordBoxAgain.Password)

s.AppendLine("Пароли отличаются");

else

if (!CheckPassword(PasswordBoxFirst.Password))

s.AppendLine("Пароль должен соотвествовать требованиям:\n - не менее 6 символов; " +

"\n - заглавные и строчные буквы;\n - не менее одного спецсимвола;\n - не менее одной цифры.");

}

return s;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

// выполняем транзакцию

using (var transaction =

MaterialsCalculatorEntities.GetContext().Database.BeginTransaction())

{

try

{

// формируем нового пользователя, добавляя данные из полей

\_currentUser.LastName = TextBoxLastName.Text;

\_currentUser.FirstName = TextBoxFirstName.Text;

\_currentUser.MiddleName = TextBoxMiddleName.Text;

\_currentUser.Email = TextBoxEmail.Text;

\_currentUser.Password = PasswordBoxAgain.Password;

\_currentUser.GenderID = 1;

if (ComboBoxGender.SelectedIndex == 1)

\_currentUser.GenderID = 2;

\_currentUser.RoleID = 2;

MaterialsCalculatorEntities.GetContext().Users.Add(\_currentUser);

MaterialsCalculatorEntities.GetContext().SaveChanges();

transaction.Commit();

// Сохраняем изменения в БД

MessageBox.Show("Запись добавлена");

DialogResult = true; // Возвращаемся на предыдущую форму

Manager.CurrentUser = \_currentUser;

}

catch (Exception ex)

{

transaction.Rollback();

MessageBox.Show(ex.Message.ToString());

}

}

}

//подбор имени файла

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

DialogResult = false;

}

/\*private void CheckBoxIsVisiblePassword\_Unchecked(object sender, RoutedEventArgs e)

{

PasswordBoxFirst.Password = TextBoxPasswordFirst.Text;

PasswordBoxFirst.Visibility = Visibility.Visible;

TextBoxPasswordFirst.Visibility = Visibility.Collapsed;

PasswordBoxAgain.Password = TextBoxPasswordAgain.Text;

PasswordBoxAgain.Visibility = Visibility.Visible;

TextBoxPasswordAgain.Visibility = Visibility.Collapsed;

}\*/

}

}

**Приложение B**

**Диск с исполняемым кодом программного модуля**